UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/809,091	03/25/2004	Christopher G. Cifra	5150-82300	7642

7590 05/04/2009

Jeffrey C. Hood
Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767

EXAMINER	
PAN, HANG	

ART UNIT	PAPER NUMBER
4123	

MAIL DATE	DELIVERY MODE
05/04/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/809,091	CIFRA, CHRISTOPHER G.	
	Examiner	Art Unit	
	HANG PAN	4123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *27 March 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *27 March 2009* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 4 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 4 recites "**The memory medium of claim 2, wherein the diagram comprises one or more control structures, wherein the one or more control structures control execution of the set of function blocks; and wherein the one or more control structures comprises one or more of: conditional branching; or looping**". In applicant's remark filed on March 27, 2009, applicant suggests that the language of the claim, namely "**wherein the one or more control structures"** coupled with **"wherein the one or more control structures *control execution of* the set of function blocks"** requires *the diagram* to actually include the control structures, rather than merely *represents* the control structures as construed by the examiner, see page 14 of applicant's remark, "… **this is not a conditional in *the diagram*, is not represented *in the diagram*, and is not exercised during execution of *the diagram* (the executable generated from *the diagram*)"** [*emphasis added*]. However, the specification states **"[0008] … In response to the user constructing *a diagram or graphical program* using the block diagram editor, data structures and/or program instructions *may be automatically constructed* which characterize an**

execution procedure that *corresponds to the displayed procedure*. The graphical program may be compiled or interpreted by a computer" and **"[0022] In one embodiment, the set of function blocks may be displayed in a diagram, e.g., in a specified display area *of the GUI***". The claim as explained by applicant's remarks, however seems to require that *the diagram* itself, i.e. the GUI to include the control structures and further that the control structures contained in the GUI subsequently control the execution of the function blocks, which has not been so described in the specification, nor does it seem technologically feasible that *the image* on a GUI actually controls the execution of function blocks. Thus the claim fails to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Nevertheless, because applicant's claim is clearly technologically deficient, the examiner continues to construe *the diagram* in the claim to model or represent a control structure (program codes), somewhere other than on the GUI, and that it is this model or representation that actually controls the execution of the set of function blocks.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 23 and 24 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. While the claims recite a series of steps or acts to be performed, a statutory "process" under 35 U.S.C. 101 must (1) it is tied to a particular machine or apparatus, or (2) it transforms a particular article into a

different state or thing (see *In re Bilski*, 88 USPQ2d 1385 (Fed. Cir. 2008)). While the preamble of the claim recites a computer is utilized to perform a process, the steps of the process are not positively tied to a particular machine. Thus the claims neither transform underlying subject matter nor positively tie to another statutory category that accomplishes the claimed method steps, and therefore do not qualify as a statutory process.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1 – 14, 16, 19-24 are rejected under 35 U.S.C. 102(e) as being anticipated by Zink et al. referred hereinafter "Zink" (US Patent 6,738,964).

With respect to claim 1: "A memory medium that stores program instructions" **Zink teaches CPU and memory medium to load the program, see Figure 1 and page 3, line 23-26;** "implementing a plurality of function blocks for use in specifying and performing a signal analysis function utilizing a plurality of instruments wherein the plurality of instruments comprises two or more virtual instruments (VIs)", **Fig. 22A in Zink shows multiple blocks(such as elements COM, ADC and DAC) comprised of**

multiple virtual instruments(such as elements COM, ADC and DAC); The above elements are termed as "blocks" or "components" by Zink(see page 27 line 7 - 20, which are equivalent to applicant's term "function block"(see Zink's definition of "block" and "component" in Page 4 Table 1); Note the applicant states a virtual instrument may be implemented in software(specification Page 4, line 27-29); "wherein each function block comprises: a function block icon operable to be displayed in a graphical user interface (GUI) of a signal analysis function development environment, wherein the function block icon visually indicates a respective signal operation", **Element ADC in Fig. 22A is displayed as an icon in a GUI environment, the icon visually indicates it can perform analog to digital operation;** "and a set of program instructions associated with the function icon, wherein the set of program instructions are executable to perform the respective signal operation", **Fig. 16B shows a list of program instruction files that are used to perform the signal operation of a functional block called signal generator in Fig. 16A(Zink page 14 line 33-41);** "wherein each function block is selectable from the plurality of function blocks by a user for inclusion in a set of function blocks, and wherein each function block operates to perform the respective signal operation continuously upon being selected", **Fig. 18 shows a selection of function blocks (Signal Generator, Filter ...) by user when creating a function block which can be used in a set of function blocks. A function block would continuously perform the function of a signal generator after being selected for signal generating function. (Page 18, line 18-28);** "wherein each function block is operable to provide a respective

output based on the respective signal operation, wherein the respective output is operable to be displayed in the GUI, provided as input to one or more other ones of the set of function blocks, or exported to an external device", **Fig. 22B shows the output of a function block in a GUI environment for the signal operation which the function block is set to perform(Zink page 21 line 50-54) Element "Side Gain" on Fig. 22A is shown to provide an output of a signal operation, this output is used as an input to another element; The interoperation between function blocks is described in Zink page 7, line 28 - 49;** "and wherein the set of function blocks is executable to perform the signal analysis function under the signal analysis function development environment using one or more of the plurality of instruments", **Fig. 22A shows a set of function blocks performing a signal analysis function using plurality of instruments(Zink page 21, line 39-53).**

Claim 2: "The memory medium of claim 1, wherein the set of function blocks are displayed in a diagram, wherein the diagram comprises one or more of: a linear sequence; a data flow diagram; a tree diagram; and a dependency diagram".
Fig. 22A in Zink shows a set of function blocks are displayed in a data flow diagram on right and a tree diagram on the left.

Claim 3: "The memory medium of claim 2, wherein the diagram substantially visually represents I/O relationships between the function blocks; and wherein, when

the I/O relationships between the function blocks change, the diagram is automatically updated in accordance with the changed I/O relationships between the function blocks". **Fig. 22A in Zink shows a data flow diagram, which indicates the I/O relationships between the function blocks (i.e. ADC's output signal is the input signal for Side Gain, indicated by the line connecting two; when the input-output order of function blocks changes, the diagram would reflect the change). The interoperation between function blocks is described in Zink page 7, line 28 – 49.**

Claim 4: "The memory medium of claim 2, wherein the diagram comprises one or more control structures, wherein the one or more control structures control execution of the set of function blocks; and wherein the one or more control structures comprises one or more of: conditional branching; or looping". **Fig. 16C in Zink shows program codes associated with a control structure "Signal Generator" in a set of function blocks; a signal generator icon can be represented on a diagram as in Fig. 17A, element 1701. The "Select Case SelectedWaveform" statement is a conditional statement that controls the execution of the function blocks. Depending on the value of the variable "SelectedWaveform", the control structure would generate different types of signal (Zink page 14, line 35 - 44).**

Claim 5: "The memory medium of claim 2, wherein the diagram comprises information specifying the respective signal operations of the set of function blocks, and

wherein the information is executable to perform the signal analysis function under the signal analysis function development environment".

The tree diagram on Fig. 22A in Zink contains a list of information (files) specifying the respective signal operations of the set of function blocks to the right. The information (files) is executable to perform signal analysis function. Fig. 16C shows an example of such information (file) (Zink page 14, line 35 – 44).

Claim 6: "The memory medium of claim 5, wherein the information specifying the respective signal operations of the set of function blocks is useable to generate a program, and wherein the program is executable to perform the signal analysis function independently of the signal analysis function development environment".

Zink page 6, line 42 - 53 state "Component assembly tool 507 also provides an interface to code-generation and other build tools 505. This facility permits the project or projects to be compiled, assembled, linked, built, or it can invoke any other build process needed. Each component includes some embedded build information. Output files 506, which are the result of the build process, can be directly loaded into appropriate target platforms, or may be manually loaded into target platforms…"

Claim 7: "The memory medium of claim 1, wherein each of at least a subset of the plurality of function blocks is operable to: receive a signal from a signal source; perform the respective signal operation on the signal; and output a result of the

respective signal operation for one or more of: display in the GUI; storage; input to another one of the plurality of function blocks; and export to an external device".

Fig. 22A in Zink shows a function block ALU on the right side receives a signal from signal source Tone Gain, it synthesizes this signal with the signal input from Side Gain to produce an output at S50. This output is displayed in Fig. 22B (Zink page 21 line 50-54). The interoperation between function blocks is described in Zink page 7, line 28 – 49.

Claim 8: "The memory medium of claim 1, wherein the set of program instructions are further executable to: receive user input selecting the function block icon; display a configuration GUI for the function block; and receive user input to the configuration GUI setting one or more parameters of the function block, thereby configuring the function block, wherein the configured function block is operable to perform the signal operation in accordance with the one or more set parameters".

Fig. 15 in Zink shows a set of program instruction running to set up the property of a filter function block. The user can select the function block as a "Band Pass" filter. The configuration of the function block is displayed in this GUI window. There are multiple parameters a user can set to affect the operation of the function block (Zink page 14, line 5 -22)

Claim 9: "The memory medium of claim 8, wherein each function block has a default configuration, wherein, prior to said configuring the function block, the function

block is operable to perform the signal operation in accordance with the default configuration". **Zink page 16 line 67 – page 17 line 4 teach a function block can perform signal operations based on a default configuration. "The first time that the property dialog window is presented, the user will have had no opportunity to make any property modifications. In this case, this section of code detects that no properties have been set, and default values for all properties are automatically generated. "**

Claim 10: "The memory medium of claim 1, wherein at least one of the plurality of function blocks comprises a user-defined function block, and wherein the set of program instructions of the user-defined function block are executable to perform a user-defined signal operation". **Zink page 13 line 14-26 teach user-defined function blocks. "User-defined components are representatives of another type of component. These are "component shells" that have little or no inherent code. User-defined components provide editing capabilities (like text editing) so that software code (e.g. C source code) can be entered directly by the user, pins can be added, and properties can be defined. These components are similar in every way to other components, but they permit a user to completely define their functionality (via user-entered code rather than graphical components). After the user finishes writing and debugging the code in a user-defined component, the user-defined component will be graphically represented in a drawing as a block similar to any other block".**

Claim 11: "The memory medium of claim 10, wherein the set of program instructions of the user-defined function block comprises a pre-defined program".

Fig. 20A in Zink teaches how to create a user defined function block using pre-defined program (existing DSP codes). Zink page 19, line 44 - 48 further describes Fig. 20A, "The graphical solutions development system 500 is based on the concept of the composition of several pre-engineered components to provide a higher-level solution".

Claim 12: "The memory medium of claim 1, wherein each function block comprises an input and an output, wherein the input is operable to receive signals from one or more of: an external signal source; a file; or another function block; and wherein the output is operable to send resultant signals to one or more of: a display of the GUI; an external device; a file; or another, different, function block".

Fig. 22A in Zink shows a set of function blocks, where each function block receives an input from a file or another function block and output signals to a display of GUI or another function block. The interoperation between function blocks is described in Zink page 7, line 28 – 49.

Claim 13: "The memory medium of claim 1, wherein each function block is operable to display respective indicators for one or both of: one or more input signals for the function block; and one or more output signals for the function block, wherein the

respective indicators comprise text or a graphical image indicating a respective signal". **Fig. 22A in Zink shows a set of function blocks. The element Side Gain contains a triangle symbol on the left side, to indicate an input signal. It also has a triangle symbol on the right side, to indicate an output signal. The triangle symbol can be an input or an output pin, as described in Zink page 7, line 7 - 15.**

Claim 14: "The memory medium of claim 13, wherein each indicator of the function block is selectable by a user to associate the respective signal with: a display of the GUI, wherein in response to being associated with the display, the respective signal is displayed in the display of the GUI; or a different function block of the set of function blocks, wherein in response to being associated with the different function block, the set of program instructions of the different function block performs the respective signal operation on the respective signal". **Fig. 22B in Zink shows a GUI display associated with the output signal triangle of the element ALU (Fig. 22A) at the position S50 (Page 21, line 39 - 47) Note Fig. 22B is a data viewer, capable of displaying the response at various points in a function block diagram (page 21, line 22 -37).**

Claim 16: "The memory medium of claim 14, wherein each function block is operable to receive user input indicating one or more input signals, and wherein the function block is operable to perform the signal operation on the indicated one or more signals in response to said user input indicating one or more input signals".

Fig. 22A in Zink shows a set of function blocks, where a user sets the element Side Gain to receive input from the element ADC and the element HiTone, as indicated by the two triangle symbols (input pins) on the left and bottom sides, and produce an output after signal operation, as indicated by the triangle symbol (output pin) on the right side. The interoperation between function blocks is described in Zink page 7, line 28 – 49.

Claim 19: "The memory medium of claim 1, wherein the two or more virtual instruments comprise at least one of: a DAQ (data acquisition) device; a digitizer; an arbitrary waveform generator; a digital I/O device; and a digital multimeter".
Fig. 22A in Zink shows a function block labeled as ADC, ADC(analog to digital converter) is considered as a digitizer device.

Claim 20: "The memory medium of claim 1, wherein one or more of the two or more VIs comprises a hardware device, and wherein the hardware device comprises at least one of: a DAQ (data acquisition) board; a digitizer board; an arbitrary waveform generator board; a digital I/O board; and a digital multimeter board". **Zink page 13, line 57-59 state "Components may also be used to represent the combined functionality of an aggregate composed of both hardware and software. An example might be a modem component". Zink's "modem component" is equivalent to applicant's claim language "a digital I/O board".**

Claim 21: "The memory medium of claim 1, wherein the plurality of instruments comprises at least one standalone hardware device". **Zink page 13, line 57-59 state "Components may also be used to represent the combined functionality of an aggregate composed of both hardware and software. An example might be a modem component". An external modem component is equivalent to applicant's claim language "one standalone hardware device".**

Claim 22 is rejected under the same rationale as claim 1.

Claim 23: "A computer-implemented method for specifying and performing a signal analysis function utilizing a plurality of instruments, wherein the plurality of instruments comprises two or more virtual instruments (VIs), the method comprising utilizing a computer to perform: receiving first user input selecting a function block from a plurality of function blocks for inclusion in a set of function blocks, wherein the function block corresponds to a respective signal operation, **(Fig. 18 in Zink shows a selection of a function block by user from a plurality of function blocks (Signal Generator, Filter ...) when creating a function block which can be used in a set of function blocks.)** and wherein the function block comprises a function block icon that visually indicates the respective signal operation and is operable to be displayed in a graphical user interface (GUI) of a signal analysis function development environment, and a set of program instructions associated with the function icon, wherein the set of program instructions are executable to perform the respective signal operation using at least one

of the two or more VIs; the function block performing the respective signal operation substantially continuously upon being selected, thereby performing at least a portion of the signal analysis function; and the function block providing a respective output based on the respective signal operation, (**A function block would continuously perform the function of a signal generator after being selected for signal generating function(Page 18, line 18-28). Fig. 22A shows a function block icon ADC which visually indicates its respective signal operation (analog to digital converter) in a GUI environment. Fig. 16B shows a list of executable program instruction files which are associated with a Signal Generator function block shown in Fig. 16A (page 14 line 33-41).)** wherein the respective output is provided for display in the GUI, provided as input to one or more other ones of the set of function blocks, and/or exported to an external device; wherein the set of function blocks is executable to perform the signal analysis function under the signal analysis function development environment using one or more of the plurality of instruments". **On Fig. 22A, the function block ALU on the right side provides an output signal to the function block DAC. The output of ALU is displayed in a GUI on Fig. 22B. Fig. 22A shows the function block DAC receives signal input from ALU on the left, and performs signal operation under a signal analysis function development environment. The interoperation between function blocks is described in Zink page 7, line 28 – 49.**

Claim 24: "The method of claim 23, wherein the program instructions are further executable to implement:" (**Fig. 15 in Zink shows an execution of program**

instructions in the configuration of a filter function block.) "receiving second user input to the function block invoking display of a configuration GUI for the function block; displaying the configuration GUI in response to said receiving second user input; (**A second user input invokes the GUI display of the configuration window. A window to the right of "Band Pass" option shows the output characteristics of the function block in response to the user's input.)** receiving third user input to the configuration GUI specifying values of one or more parameters of the function block, thereby configuring the function block; and the function block performing the signal operation in accordance with the one or more parameters". **A third user input of a value at the F1 field can further configure the function block. The function block can perform a signal operation in accordance to the user input parameters (Zink page 14, line 9 -22).**

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 15, 17 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zink et al. referred hereinafter "Zink" (US Patent 6,738,964), in view of Austin (US Patent Publication 2002/0070966).

As per claim 15, Zink teaches the memory medium of claim 14, wherein, in being selectable by a user, each indicator of a signal is operable to display the respective signal on the display of the GUI. Zink does not teach each indicator is operable to display the respective signal on the display of the GUI by dragged and dropped onto the display of the GUI. Austin teaches a method of associating two elements in GUI by dragging one element(such as icon) and dropping to another element(diagram window) (Austin page 21, line 39-50). It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Zink such each indicator of an input or an output signal is operable to a display of respective signal on the display of the GUI by dragging the indicator and dropping to the display of the GUI because this method allows user easily to interface with various types of data sources(such as signal indicators) and targets(such as graphs) (Austin page 24, line 26 -39)

As per claim 17, Zink teaches the memory medium of claim 16, ink does not teach user input indicating one or more input signals comprises: the user dragging and dropping one or more signal icons onto the function block. Austin teaches a method of associating two elements in GUI by dragging one element (such as an icon) and dropping to another block element (see Austin page 6, line 39-48). It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Zink such the user input indicating one or more input signals comprises: the user dragging and dropping one or more signal

icons onto the function block because this method allows user easily to interface with various types of data sources(such an signal icons) and targets(such as function blocks) (Austin page 24, line 26 -39).

As per claim 18, Zink teaches the memory medium of claim 16, Zink does not teach wherein said user input indicating one or more input signals comprises: the user selecting at least one signal in the GUI display; and the user dragging and dropping a corresponding at least one signal icon from the graph onto the function block, wherein the at least one signal icon represents the at least one signal in the GUI display. Austin teaches a method of associating two elements in GUI by dragging one element (such as icon) and dropping to another block element (see Austin page 6, line 39-48). It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Zink such the user input indicating one or more input signals comprises: the user selecting at least one signal in the GUI display; and the user dragging and dropping a corresponding at least one signal icon from the graph onto the function block, wherein the at least one signal icon represents the at least one signal in the GUI display because this method allows user easily to interface with various types of data sources(such as signal icons) and targets(such as function blocks) (Austin page 24, line 26 -39).

Response to Arguments

Applicant's arguments filed on March 27, 2009 have been fully considered but they are not persuasive.

A) In response to applicant's argument on page 13 pertaining to claims 1, 22 and 23 that cites:

> **Nowhere does the cited art teach wherein each function block operates to perform the respective signal operation continuously upon being selected, as recited in claim 1.**
>
> **Cited Figure 18 and related text disclose components for Zink's block diagrams, where blocks representing the components are connected with wires to graphically implement programs for digital signal processors and system designs. However, Zink's block diagrams are included in projects that are "compiled, assembled, linked, built" (see, col.6:42-45). The resulting executable may then be deployed to target platforms for execution. For example, col.11:61-63 states "Platform components are components that contain information about the target hardware where the project's executable code will be 'run'". Nowhere does Zink mention or even hint at the components performing their respective functions continuously upon being selected, i.e., executing continuously as soon as they are added to the block diagram.**
>
> **Thus, the cited art fails to disclose this feature of claim 1.**
>
> **Thus, for at least this reason, Applicant submits that the cited art fails to teach or suggest all of the features and limitations of claim 1. Thus claim 1, and those claims respectively dependent therefrom, are patently distinct and nonobvious over the cited art, and thus allowable.**
>
> **Independent claims 22 and 23 include similar limitations as claim 1, and so the above arguments apply with equal force to these claims. Dusk, for at least the above reasons, claims 22 and 23, and those claims respectively dependent therefrom, are similarly patently distinct and nonobvious over the cited art, and thus allowable**.

Examiner respectfully disagrees.

Applicant argues **"Zink's block diagrams are included in projects that are "compiled, assembled, linked, built" (see, col.6:42-45). The resulting executable**

may then be deployed to target platforms for execution". However, Zink discloses **"Besides offering the user a graphical drawing environment, component assembly tool 507 provides an extended user interface that permits the user to build, load, run, and debug complex projects**" (Page 6, line 21-24), meaning the graphic development system is capable of running a project containing function blocks, a project does not necessarily have to be deployed to a target platform for execution. Fig. 22B is an example of an output of a function block during the execution on the graphic development system. Applicant also argues "**Nowhere does Zink mention or even hint at the components performing their respective functions continuously upon being selected, i.e., executing continuously as soon as they are added to the block diagram"**. The claim does not recite that the function block begins operation "**as soon as they are added**" as argued. Examiner construes the claim limitation of "**wherein each function block operates to perform the respective signal operation continuously upon being selected**" to be interpreted as "after each function block is selected to perform a signal operation, it will perform the signal operation continuously, until it is terminated". Fig 22B is a scope view showing the operation of a function block in a diagram; for example, if the scope view is attached to a signal generator, the scope view would show continuously the operation of the signal generator, until the operation is terminated. The action of selecting a function block may involve several steps.

B) In response to applicant's argument on page 14 pertaining to claim 4 that cites:

> **For example, nowhere does the cited art teach wherein the diagram comprises one or more control structures,**

wherein the one or more control structures control execution of the set of function blocks; and wherein the one or more control structures comprises one or more of: conditional branching; or looping, as recited in claim 4.

Cited Figure 16C illustrates multiple code modules in a development component, where, as the related text explains, the particular code module included in the project depends on property settings. Applicant respectfully notes that the citation does not disclose control structures, i.e., conditional branching or looping, in the diagram. More specifically, note that the cited "conditional" simply refers to the fact that the user configures a property, e.g., "triangle waveform", which results in inclusion of the appropriate code module, e.g., "the 'triangle files' (triangle.h, triangle.c, and triangle.obj) into the project". Clearly, this is not a conditional in the diagram, is not represented in the diagram, and is not exercised during execution of the diagram (the executable generated from the diagram).

Thus, the cited art fails to disclose this feature of claim 4.

Examiner respectfully disagrees.

Applicant argues **Figure 16C "does not disclose control structures, i.e., conditional branching or looping, in the diagram**". Applicant is interpreting control structures on the diagram (the graphic pixels on the display) control the execution of the function blocks. A person of ordinary skill in the art would interpret the control structures on the diagram as mere graphic representations of the associated program codes. The control structures are displayed for the benefit of the users. What controls the execution of the function blocks is not the graphic representation of a control structure, but the associated program codes which the control structure represents. Figure 16C shows program codes associated with a control structure "signal generator", the signal generator may be represented on a diagram similar to element 1701 of Fig. 17A.

Applicant also argues **"Clearly, this is not a conditional in the diagram, is not**

represented in the diagram, and is not exercised during execution of the diagram". The program codes in Fig. 16C are represented by a control structure "signal generator" in a diagram, the codes contain a conditional statement "select … case …". The condition for the branching operation depends on a user input parameter. As mentioned in the response to the argument A), the graphic development system is capable of running a project containing block diagrams.

C) In response to applicant's argument on pages 14-15 pertaining to claim 5 that cites:

> **Nor does the cited art teach wherein the diagram comprises information specifying the respective signal operations of the set of function blocks, and wherein the information is executable to perform the signal analysis function under the signal analysis function development environment, as recited in claim 4 [sic, 5].**
>
> **As noted above, cited Figure 16C illustrates multiple code modules in a development component, where, per the related text, the particular code module included in the project depends on property settings. As mentioned above, Zink's projects are "compiled, assembled, linked, built" (see, col.6:42-45), where the resulting executable is deployed to target platforms for execution. Nowhere does Zink indicate that Zink's executable files perform their functionality under the development environment in which they were developed. Applicant notes that this feature of claim 4 [sic, 5] facilitates the "continuous" execution functionality discussed above with respect to claim 1, in that the function blocks can execute during development of the signal analysis function.**
>
> **Thus, the cited art fails to disclose this feature of claim 5**.

Examiner respectfully disagrees.

Applicant argues:

> **Nowhere does Zink indicate that Zink's executable files perform their functionality under the development environment in which they were developed. Applicant notes**

that this feature of claim 5 [sic, 5] facilitates the "continuous" execution functionality discussed above with respect to claim 1, in that the function blocks can execute during development of the signal analysis function. Thus, the cited art fails to disclose this feature of claim 5.

Zink discloses "**Besides offering the user a graphical drawing environment, component assembly tool 507 provides an extended user interface that permits the user to build, load, run, and debug complex projects**" (Page 6, line 21-24), meaning the graphic development system is capable of running a project containing function blocks, a project does not necessarily have to be deployed to a target platform for execution. Fig. 22B is an example of an output of a function block during the execution on the graphic development system. Furthermore, claim 5 does not describe "**facilitates continuous execution functionality**" as argued by the applicant.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to HANG PAN whose telephone number is (571)270-7667. The examiner can normally be reached on Mon-Fri 8AM-5PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, David Robertson can be reached on 571-272-4186. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/H.P./
HANG PAN
Examiner
Art Unit 4123

/David L. Robertson/
Supervisory Patent Examiner
Art Unit 4123